Design and Implementation of a Parallel Multivariate Ensemble Kalman Filter for the Poseidon Ocean General Circulation Model

Christian L. Keppenne
Science Applications International Corporation, Beltsville, Maryland

Michele M. Rienecker
Oceans and Ice Branch
NASA Seasonal-to-Interannual Prediction Project
Goddard Space Flight Center, Greenbelt, Maryland

Abstract

A multivariate ensemble Kalman filter (MvEnKF) implemented on a massively parallel computer architecture has been implemented for the Poseidon ocean circulation model and tested with a Pacific Basin model configuration. There are about two million prognostic state-vector variables. Parallelism for the data assimilation step is achieved by regionalization of the background-error covariances that are calculated from the phase-space distribution of the ensemble. Each processing element (PE) collects elements of a matrix measurement functional from nearby PEs. To avoid the introduction of spurious long-range covariances associated with finite ensemble sizes, the background-error covariances are given compact support by means of a Hadamard (element by element) product with a three-dimensional canonical correlation function.

The methodology and the MvEnKF configuration are discussed. It is shown that the regionalization of the background covariances has a negligible impact on the quality of the analyses. The parallel algorithm is very efficient for large numbers of observations but does not scale well beyond 100 PEs at the current model resolution. On a platform with distributed memory, memory rather than speed is the limiting factor.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background and motivation

This report summarizes the progress made by the NASA Seasonal-to-Interannual Prediction Project (NSIPP) at the Goddard Space Flight Center in its use of a multivariate ensemble Kalman filter (MvEnKF) to assimilate observations into the Poseidon isopycnal ocean general circulation model (OGCM) (Schopf and Loughe 1995; Konchady *et al.* 1998; Yang *et al.* 1999). NSIPP uses a coupled ocean/land/atmosphere/ice model to produce forecasts of El Niño and its global teleconnections. The coupled model's components are Poseidon, the NSIPP-1 atmospheric general circulation model (AGCM) (Suarez and Takacs 1995; Schaffer and Suarez 1998; Bacmeister and Suarez 2001), the Mosaic land-surface model (Koster and Suarez 1996) and a thermodynamic sea-ice model. A CRAY T3E is used for ensemble integrations of the parallel versions of the ocean, atmosphere and land models.

At present, a univariate form of optimal interpolation (univariate OI: UOI) is used for the ocean analyses resulting in the initial ocean state for the coupled forecasts. The UOI processes temperature measurements from the Tropical Ocean and Atmosphere (TAO, *e.g.*, McPhaden *et al.* 1998) array in the Tropical Pacific Ocean. Like several other ocean data assimilation systems currently in use at other institutions (*e.g.*, Ji and Leetma 1997), it is based on the assumption that the forecast-error covariances are approximately Gaussian and that the covariances between the temperature-field errors and the salinity-field and current-field errors are negligible.

Largely due to the high-resolution coverage and accuracy of the TAO measurements, the UOI appears to be effective in improving surface and sub-surface temperature field estimates in the equatorial region in comparison with the estimates obtained without temperature assimilation. As a result, the introduction of the UOI into the coupled forecasting system has resulted in significant improvements in the coupled model's hindcast skill of Niño-3 temperature anomalies.

The UOI has the advantage of being inexpensive in terms of computing resources. Its other main advantage is that it was relatively easy to implement within the framework of the parallel OGCM. Nevertheless, the UOI suffers from three major shortcomings. The first shortcoming is that it can only be used to assimilate measurements of a model prognostic variable. The second shortcoming is that it does not use any statistical information about the expected inhomogeneous distribution of model errors. The third shortcoming is that it is based on a steady state error-covariance model which gives the same weight to a unit innovation regardless of how accurate the ocean-state estimate has become as a result of previous analyses. Directly linked to this shortcoming is the failure to provide time-dependent estimates of the model errors.

In response to the first two shortcomings, a parallel multivariate OI (MvOI) system has been implemented. The MvOI uses steady state estimates of the model-error statistics computed from ensemble runs of the OGCM in the presence of stochastic atmospheric forcing fields. Yet, the MvOI cannot adjust to dynamically evolving error statistics. The development of a parallel MvEnKF has been undertaken to address this shortcoming.

1.2 Overview of the ensemble Kalman filter

Although the Kalman filter (Kalman 1960) and its generalization to nonlinear systems, the extended Kalman filter, are statistically optimal sequential estimation procedures that minimize error variance (Daley 1991; Ghil and Malanotte-Rizzoli 1991; Bennett 1992; Robinson *et al.* 1998), they cannot be used in the context of a high-resolution ocean or atmospheric model because of the prohibitive cost of time stepping the model-error covariance matrix when the model has more than a few thousand state variables. Therefore, reduced-rank (*e.g.*, Cane *et al.* 1996, Verlaan and Heemink 1997) and asymptotic (e.g., Fukumori and Malanotte-Rizzoli 1995) Kalman filters have been proposed. Evensen (1994) introduced the ensemble Kalman filter (EnKF) as a Monte Carlo-based alternative to the traditional Kalman filter. In the EnKF, an ensemble of model trajectories is integrated and the statistics of the ensemble are used to estimate the model errors. Closely related to the EnKF are the singular evolutive extended Kalman filter (Pham *et al.* 1998) and the error-subspace statistical estimation algorithms described in Lermusiaux and Robinson (1999).

Evensen (1994) compared the EnKF to the extended Kalman filter in twin assimilation experiments involving a two-layer quasigeostrophic (QG) ocean model on a square 17×17 grid. Evensen and van Leeuwen (1996) used the EnKF to process GEOSAT altimeter data into a two-layer, regional QG model of the Agulhas current on a 51×65 grid.

Houtekamer and Mitchell (1998) introduced a version of the EnKF in which two ensembles are integrated and—in order to maintain a representative ensemble spread when the model is assumed perfect—the statistics of each ensemble are used to update the other. They tested this algorithm in identical-twin experiments involving a three-level, spectral QG model at triangular truncation T21. In Mitchell and Houtekamer (2000), simulated radiosonde profiles were assimilated into the same model using an EnKF algorithm involving parameterized model errors.

Keppenne (2000, hereafter K00) conducted twin experiments with a parallel MvEnKF algorithm in the context of an imperfect model and parameterized model errors. The algorithm was applied to the assimilation of synthetic altimetry measurements into a two-layer, spectral, T100 primitive equation model. The state-vector size was small enough in this application to justify a parallelization scheme in which each ensemble member resides in the memory of a separate CRAY T3E processor (hereafter processing element: PE). To parallelize the analysis, K00's algorithm transposes the ensemble across PEs at analysis time, so that each PE ends up processing data from a sub-region of the model domain. The influence of each observation is weighted according to the distance between that observation and the center of each PE region.

To filter out noise associated with small ensemble sizes, Houtekamer and Mitchell (2001) developed a parallel EnKF analysis algorithm that applies a Hadamard (element by element) product (*e.g.*, Horn and Johnson 1991) of a correlation function having local compact support with the background-error covariances. They tested this analysis scheme on a 128×64 Gaussian grid corresponding to a three-level QG model using randomly generated ensembles of first-guess fields computed ahead of time, rather than a dynamically evolving ensemble of model trajectories. The benefits of constraining the covariances between ensemble members using a

Hadamard product with a locally supported correlation function has also been investigated by Hamill and Snyder (2000) in the context of an intermediate QG atmospheric model.

In this paper, we build upon the contributions made by each of the above-mentioned studies to implement a parallel MvEnKF for the Poseidon OGCM. Initial tests are undertaken with a 20-layer, Pacific basin configuration of the model with about two million state variables. The system noise is accounted for in a manner similar to that used in K00, by including a stochastic component in the forcing fields. Following Houtekamer and Mitchell (2000), the background-error covariances are multiplied element-by-element by an idealized three-dimensional compactly supported correlation function.

1.3 Organization of the following Sections

The remainder of this paper is concerned with describing the parallel MvEnKF implementation for the Poseidon model. The model is briefly discussed in Section 2 and the algorithms are presented in Section 3 where the focus is on the aspects of this EnKF implementation that differ from other implementations. To illustrate the plausibility of using the MvEnKF in an operational framework, some timing numbers are given in Section 4. The scalability of the algorithms and the effect of distributing the analysis calculations between PEs are also discussed in Section 4. Section 5 contains a summary. In a companion article (Keppenne and Rienecker 2001, hereafter KR01), the parallel MvEnKF is validated in the context of TAO-temperature and TOPEX-altimeter data assimilation and is compared with the UOI presently used quasi-operationally at NSIPP.

2 The Poseidon parallel ocean model

2.1 Model summary

The Poseidon model (Schopf and Loughe, 1995) is a finite-difference reduced-gravity ocean model which uses a generalized vertical coordinate designed to represent a turbulent, well-mixed surface layer and nearly isopycnal deeper layers. Coastal topography is represented, but the reduced-gravity treatment precludes the use of variable bottom depth. Poseidon has been documented and validated in hindcast studies of El Niño (Schopf and Loughe 1995) and has since been updated to include prognostic salinity (*e.g.*, Yang *et al.* 1999). More recently, the model has been used in an investigation of the annual cycle in the eastern Equatorial Pacific (Yu *et al.* 1997) and in a numerical study of the surface heat balance along the equator (Borovikov *et al.* 2001).

Poseidon's prognostic variables are layer thickness, $h(\lambda, \theta, \zeta, t)$, temperature, $T(\lambda, \theta, \zeta, t)$, salinity, $S(\lambda, \theta, \zeta, t)$, and the zonal and meridional current components, $u(\lambda, \theta, \zeta, t)$ and $v(\lambda, \theta, \zeta, t)$, where λ is longitude, θ latitude, t time and ζ is a generalized vertical coordinate which is 0 at the surface and increments by 1 between successive layer interfaces.

Explicit detail of the model, its vertical coordinate representation and its discretization are provided in Schopf and Loughe (1995) and are only summarized here. The equation for mass continuity is

$$\frac{\partial h}{\partial t} + \nabla.(\mathbf{v}h) + \frac{\partial w_e}{\partial \zeta} = 0, \qquad (1)$$

where $\nabla.$ and $\mathbf{v}$ are the two-dimensional (2D) divergence operator and velocity vector and w_e represents the volume flux across layer interfaces, including freshwater flux through the surface.

The heat equation is

$$\frac{\partial hT}{\partial t} + \nabla.(\mathbf{v}hT) + \frac{\partial w_e T}{\partial \zeta} = \frac{\partial}{\partial \zeta}\left(\frac{\kappa}{h}\frac{\partial T}{\partial \zeta}\right) + \frac{\partial Q}{\partial \zeta} + hF_h(T), \qquad (2)$$

where Q is the external heat flux, κ is a heat diffusivity and F_h is a 2D smoothing operator. The salinity equation is

$$\frac{\partial hS}{\partial t} + \nabla.(\mathbf{v}hS) + \frac{\partial w_e S}{\partial \zeta} = \frac{\partial}{\partial \zeta}\left(\frac{\kappa_s}{h}\frac{\partial S}{\partial \zeta}\right) + hF_h(S), \qquad (3)$$

where κ_s is a salinity diffusivity. The 2D momentum equation is

$$\frac{\partial(\mathbf{v}h)}{\partial t} + \nabla.(\mathbf{v}h\mathbf{v}) + \frac{\partial w_e \mathbf{v}}{\partial \zeta} = -\frac{h}{\rho_0}\nabla p' - bh\nabla z - f\,\mathbf{k}\times\mathbf{v} + \frac{\partial}{\partial \zeta}\left(\frac{\upsilon}{h}\frac{\partial \mathbf{v}}{\partial \zeta}\right) + \frac{1}{\rho_0}\frac{\partial \tau}{\partial \zeta} + hF_v(\mathbf{v}), \qquad (4)$$

where υ is a vertical friction, τ is the vertical shear stress, $f\,\mathbf{k}\times\mathbf{v}$ is the Coriolis acceleration and F_v is a dissipation term. A hydrostatic Boussinesq approximation is made, whereby $p'(z)$ is the pressure anomaly at depth z, b is buoyancy and ρ_o is the mean density. The hydrostatic equation then becomes

$$\frac{\partial p'}{\partial \zeta} = -\rho_0 bh. \qquad (5)$$

Following Pacanowski and Philander (1981), vertical mixing is parameterized through a Richardson number-dependent mixing scheme implemented implicitly. An explicit mixed layer is included with a mixed layer entrainment parameterization following Niiler and Kraus (1977).

A time-splitting integration scheme is used whereby the hydrodynamics are done with a short time step (15 minutes), but the vertical diffusion, convective adjustment and filtering are done with coarser time resolution (half-daily).

2.2 Model setup

The version of Poseidon used here has been parallelized as in Konchady *et al.* (1998) using the same message-passing protocol and 2D horizontal domain decomposition used by Schaffer and Suarez (1998) for the AGCM.

The experiments described in this article use a 20-layer Pacific basin version of the parallel model with uniform 1º zonal resolution. The meridional resolution varies between 1/3º at the equator and 1º in the extratropics. A solid boundary is imposed at 45º south. There, a no-slip condition is used for the currents and a no-flux condition is used for mass, heat and salinity. The issue of the forcing is discussed in Section 3.12.

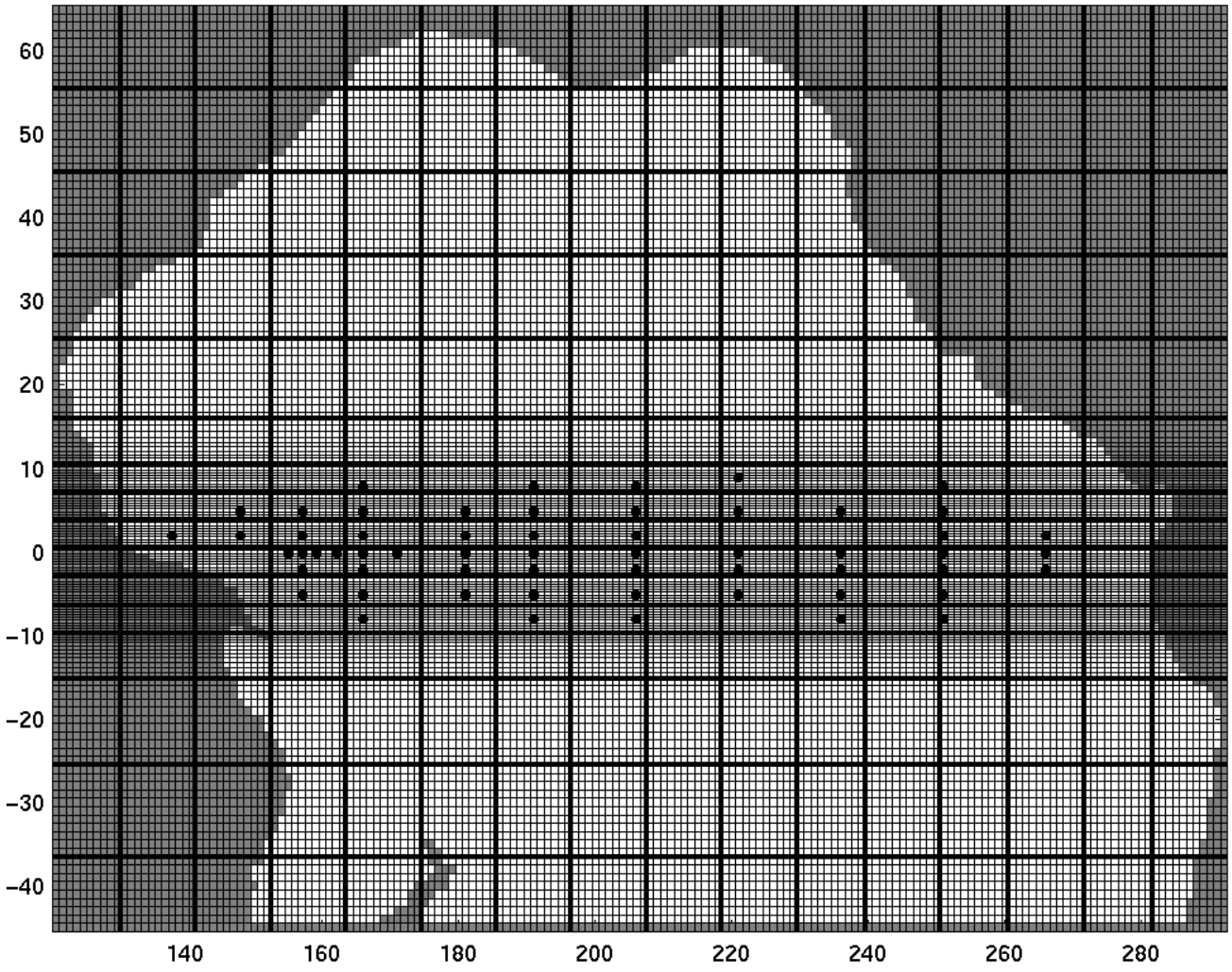

Figure 1. Horizontal domain decomposition for the Pacific model. The thin lines delineate grid cells. The thick lines correspond to the boundaries of each PE box on the 16×16 PE lattice. Each dark circle corresponds to a TAO mooring.

There are $173 \times 164 \times 20$ grid boxes, of which 28% are situated over land, resulting in a total of 2.0422×10^6 individual prognostic variables. A 16×16 PE lattice is used as shown in Figure 1. The PEs located over land are virtual PEs which do not take part in the ensemble integrations and analyses.

Figure 2 illustrates the horizontal setup for one PE box. Locally within the box, the grid cells are numbered $1 \le i \le I$, zonally and $1 \le j \le J$, meridionally, from the box's lower-left, southwest corner. In order to minimize the communication overhead in the horizontal differencing of the model equations, the PE boxes overlap. The overlapping regions, called halo regions, have width $i_1 - 1$ to the West, $I - i_2$ to the East, $j_1 - 1$ to the South and $J - j_2$ to the North. The PE-private regions are thus defined by $i_1 \le i \le i_2$ and $j_1 \le j \le j_2$. Vertically within each grid cell, the grid boxes are numbered $1 \le k \le K$.

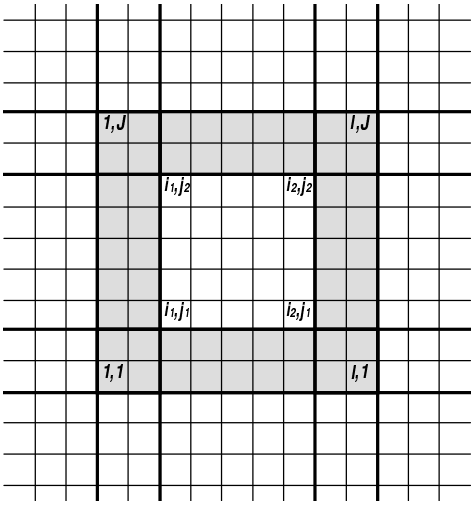

Figure 2. Schematic setup for one PE. The halo regions are colored gray. The thin lines delineate grid cells. The thick lines delimit the halo regions and PE boundaries. In this example, $I = J = 9$, $i_1 = j_1 = 3$ and $i_2 = j_2 = 7$.

3 Assimilation methodology

3.1 Horizontal domain decomposition

In K00, the number of model state variables, 2.7×10^5, was small enough to integrate each ensemble member on a separate PE. Across-PE transpositions of the ensemble were used to conduct the analyses in parallel (Fig. 1 in K00). After each transposition, each PE contained the state-vector elements of every ensemble member that correspond to a sub-region of the model domain, rather than the entire state of a single ensemble member as it did before the transposition. Local background covariances were computed on every PE from the local ensemble distribution and were then used to calculate the analysis increments.

An advantage of K00's algorithm is that both the error-covariance forecasting step (ensemble integration) and the analysis step occur in parallel, although the model itself is coded serially.

Yet, there are two obvious disadvantages to this scheme. First, it can only be used if each copy of the model can fit in the memory of a single PE. This precludes using this algorithm with a high-resolution GCM on most massively parallel processors (MPPs) with distributed memory. On such computer architectures, the memory of each PE is usually insufficient to contain the entire state vector of a GCM. Second, the ensemble transpositions across PEs involve a significant communication overhead.

Since the version of Poseidon used here is parallelized, the same domain decomposition used to run the model can be used in the analyses, provided the background error-covariance matrix, $\boldsymbol{P}^f$, is locally approximated. This simplification avoids costly ensemble transpositions across PEs. Thus, the ensemble is distributed so that the memory of each PE contains the same elements of each ensemble member's state vector. These elements correspond to every variable contained within the PE boxes illustrated in Figure 2. This decomposition is used for the ensemble integrations as well as for the analyses.

3.2 Assimilation on geopotential surfaces

The temperature measurements from each TAO mooring are recorded at specific depths which are fairly consistent between moorings. Since Poseidon uses an isopycnal vertical coordinate, the model fields must be interpolated to the latitude, longitude and depth of each observation. With the MvEnKF, the elements of $\boldsymbol{P}^f$ can be calculated in the (λ, θ, ζ) coordinate system and the analysis can occur on isopycnals, whereby the vertical interpolation can be made part of the measurement functional (Section 3.9). To the contrary, when the UOI was implemented, the choice was made to treat the temperature observations in the usual (λ, θ, z) coordinate system in light of the absence of corresponding salinity observations. Thus, to maintain compatibility with the UOI which interpolates model fields vertically to a series of pre-specified depths (hereafter levels) prior to each analysis, the same approach is used here and the background covariances are calculated on levels rather than on layers[1]. Therefore, the T, S, u and v fields are converted from isopycnals to levels and the analysis increments are calculated on the levels before being mapped back to the isopycnals. Sixteen levels are used in KR01.

The above scheme results in only T, S, u and v being updated. The layer thicknesses, h, are left unchanged by the assimilation. Rather, the procedure allows the model to dynamically recalculate h from to the new density distribution and the target interface buoyancies, as it does at every time step (see Schopf and Loughe 1995). Thus, the decision not to calculate h increments is deliberate. Rather, the incremental update mechanism discussed in Section 3.8 lets the model dynamically adjust the layer thicknesses using the information contained in the T, S, u and v increments.

Since only the layer-average value of T, S, u and v in each grid box (i, j, l) appear in the model equations, the mapping from isopycnals to levels could be made by assigning to a given field at (λ_{ij}, θ_{ij}, z_k).the value of the same field at (λ_{ij}, θ_{ij}, l). However, if the mapping were performed in this manner, ambiguities would arise when several levels pass through the same layer at (λ_{ij}, θ_{ij}).

[1] The latest version of the NSIPP ocean data assimilation code implemented after running the experiments discussed in this study allows the user to choose between mapping the model state to levels prior to each analysis or conducting the assimilation on the quasi-isopycnal layers.

A possible consequence is the singularity of the analysis equations of Section 3.6 in the (λ, θ, z) coordinate system. To avoid this problem, the mapping is made as though the vertical variations of the field were piecewise linear, with the discontinuities in the slope occurring in the middle of the layers. This is illustrated in Figure 3 for the temperature field.

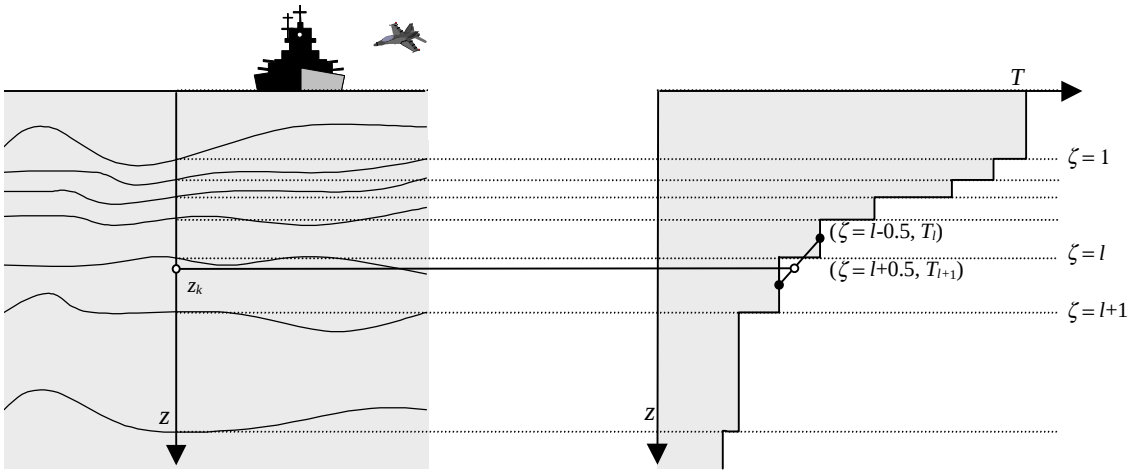

Fig.ure 3. Mapping of the model temperature field to a specified level, $z = z_k$. Within the current grid cell, z_k is contained between the layer interfaces $\zeta = l$ and $\zeta = l+1$. In the model discretization, only the layer-average temperature matters. Yet, to avoid ambiguities when more than one specified level pass through the same layer in the grid cell, the field is interpolated linearly as shown.

3.3 Ensemble size

With the MvEnKF, PE memory imposes constraints on both the domain decomposition and the ensemble size. With the usual compromise between parallelism and communication, the Pacific basin version of Poseidon is typically run on 64 PEs. The goal is for the MvEnKF runs to be done on a few times as many PEs. In this study, 256 PEs are used and the memory available on these PEs imposes a limit of about 40 ensemble members. Encouraging results have been obtained with comparably sized ensembles by Mitchell and Houtekamer (2000) with a three-level QG model and by K00 with a two-layer shallow water model. Moreover, a common objective of ocean and atmospheric modelers when they gain access to more powerful computer systems is to increase their model resolution. Therefore, it is sensible to expect that the largest ensembles of GCMs that one will be able to run concurrently on a single MPP will generally remain on the order of a few tens. In order to demonstrate that the MvEnKF can be used in a quasi-operational setting with a high-resolution GCM, one thus has to show that it can perform as well as the simpler methods currently in use at most centers, even with as few as 40 ensemble members.

3.4 Compactly supported covariances

The small ensemble size considered here introduces the need to filter out spurious long-range correlations when the background covariances are computed. Following Houtekamer and Mitchell (2000) and a suggestion by Gaspari and Cohn (1999), this filtering is achieved through a Hadamard product (*i.e.* $\boldsymbol{A} \bullet \boldsymbol{B}$ such that $\{\boldsymbol{A} \bullet \boldsymbol{B}\}_{ij} = \boldsymbol{A}_{ij} \boldsymbol{B}_{ij}$) of the error-covariance matrices with a local compactly supported correlation function.

The compactly supported correlation function is the product of a horizontal correlation function, $C_h(r_h^{(12)})$, $r_h^{(12)} = [(\lambda_2 - \lambda_1)^2/l_\lambda + (\theta_2 - \theta_1)^2/l_\theta]^{0.5}$, and a vertical correlation function, $C_v(r_v^{(12)})$, $r_v^{(12)} = |z_2 - z_1|/l_z$, where $(\lambda_i, \theta_i, z_i)$ are the coordinates of point i. In this study, $C_h = C_v = C_0$, where C_0 is defined by (4.10) of Gaspari and Cohn (1999). The normalization is such that $C_0(r) = 0, \quad r \geq 2$. The correlation scales are $l_\lambda = 30^o$, $l_\theta = 15^o$ and $l_z = 500$m for the assimilation of TAO temperature data. These scales are slightly broader than those used to define the univariate idealized correlation function employed in the UOI. When gridded TOPEX altimeter data are assimilated, $l_\lambda = 15^o$ and $l_\theta = 7.5^o$. These shorter correlation scales give better results than longer ones given the high horizontal coverage of the altimeter data.

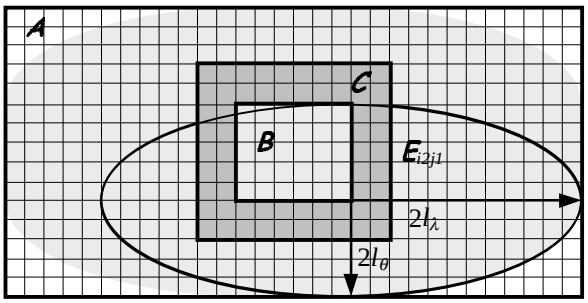

Figure 4. Domain decomposition for the analysis. The outer rectangle delimits the area, $\boldsymbol{A}$, from which the data assimilated on one PE are collected. The innermost rectangle depicts the boundary of the PE-private area (grid cells (i, j) with $i_1 \leq i \leq i_2, \quad j_1 \leq j \leq j_2$), $\boldsymbol{B}$. The ellipse delimits the influence region, $\boldsymbol{E}_{i2j1}$, of the PE-private area's southeastern corner cell, (i_2, j_1). The shaded area contains the ellipses, $\boldsymbol{E}_{ij}$, for all grid cells, (i, j), contained in $\boldsymbol{B}$. The region $\boldsymbol{C}$ contains all the PE's grid cells ($1 \leq i \leq I, \quad 1 \leq j \leq J$), including the halo regions.

3.5 Confined analysis

Although the TAO temperature data assimilated here are sufficiently few (about 600 at each analysis) for each PE to process them all, an approach whereby each PE processes data from a sub-region of the model domain is used. Besides the obvious efficiency gain in a parallel environment, another justification for this approach is that the compactly supported background covariances result in the data that directly (*i.e.*, through the measurement functional of Section 3.8) influence the state variables within each grid cell being contained within an ellipse with semi axes $2l_\lambda$ and $2l_\theta$. Taking advantage of this fact, the region from which the observations

assimilated on each PE are collected is chosen to be the smallest rectangle, with sides λ_{i2j1} - λ_{i1j1} + $4l_\lambda$ and θ_{i1j2} - θ_{i1j1} + $4l_\theta$, containing all the ellipses that correspond to the PE-private grid cells of this PE. This is illustrated in Figure 4.

3.6 Analysis equations

Without the Hadamard product of the background-error covariances with the compactly supported correlation function, the EnKF analysis equations can be written as

$$\boldsymbol{y}_i = \Xi(\boldsymbol{x}_i^f - \langle \boldsymbol{x} \rangle^f), \qquad (6a)$$

$$\boldsymbol{l}_i = \mathcal{L}(\boldsymbol{y}_i + \langle \boldsymbol{x} \rangle^f) - \mathcal{L}(\langle \boldsymbol{x} \rangle^f), \qquad (6b)$$

$$\boldsymbol{Y} = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_m\}, \quad \boldsymbol{L} = \{\boldsymbol{l}_1, \cdots, \boldsymbol{l}_m\},$$

$$\left[\boldsymbol{L}\boldsymbol{L}^T + \boldsymbol{W}\right]\boldsymbol{b}_i = \boldsymbol{d} - \mathcal{L}(\boldsymbol{y}_i + \langle \boldsymbol{x} \rangle^f) + \boldsymbol{e}_i, \qquad (6c)$$

$$\boldsymbol{x}_i^a = \boldsymbol{x}_i^f + \boldsymbol{Y}\boldsymbol{L}^T\boldsymbol{b}_i. \qquad (6d)$$

In (6) and throughout this discussion, uppercase boldface symbols represent matrices, lowercase boldface symbols represent vectors and lowercase regular (*i.e.*, not bold) symbols denote scalar variables. Boldface subscripts, such as in $\boldsymbol{x}_i$, identify the *i*th instance of the $\boldsymbol{x}$ vector. Regular subscripts identify array elements. The vector, $\boldsymbol{d}$ ($n_d \times 1$) contains n_d observations, $\boldsymbol{x}_i$ ($n_x \times 1$), $1 \le i \le m$, is the *i*th ensemble state vector and *m* stands for the ensemble size. The superscripts ^{a} and ^{f} refer to the analyzed state and the forecast, respectively, Ξ is a smoothing operator (Section 3.11) and < > denotes an ensemble average. The vectors $\boldsymbol{y}_i$ ($n_x \times 1$) and $\boldsymbol{l}_i$ ($n_d \times 1$) are columns of the matrices $\boldsymbol{Y}$ ($n_x \times m$) and $\boldsymbol{L}$ ($n_d \times m$) respectively, and $\mathcal{L}(\boldsymbol{x})$ is a measurement functional which relates the state vector to the observations (Section 3.9). Matrix $\boldsymbol{W}$ ($n_d \times n_d$) is the measurement-error covariance matrix. The representer matrix, $\boldsymbol{R} = \boldsymbol{L}\,\boldsymbol{L}^T$, maps the background-error covariance matrix, $\boldsymbol{P}^f$ ($n_x \times n_x$), to the error subspace of the measurements. The elements of $\boldsymbol{b}_i$ are the representer-function amplitudes used to update $\boldsymbol{x}_i$. The $n_d \times 1$ vector, $\boldsymbol{z}_i = \boldsymbol{d} - \mathcal{L}(\boldsymbol{y}_i + \langle \boldsymbol{x} \rangle^f) + \boldsymbol{e}_i$, contains the innovations with respect to the *i*th ensemble member. Prior to their calculation, Ξ is applied to smooth $\boldsymbol{x}_i$. Following Burgers *et al.* (1998), $\boldsymbol{e}_i$ is a random perturbation chosen such that $\langle \boldsymbol{e}_i \rangle = 0$ and $\left\langle \boldsymbol{e}_i \boldsymbol{e}_i^T \right\rangle = \boldsymbol{W}$. Its role is to maintain the influence of observation uncertainty in the error covariances estimated directly from the ensemble so that these covariances are consistent with the theoretical estimates. Its inclusion helps prevent the ensemble from collapsing resulting in a systematic error underestimation. The introduction of this term is crucial to maintain a representative ensemble variance when the matrix norm of $\boldsymbol{W}$, $|\boldsymbol{W}|$, is comparable to or greater than $|\boldsymbol{R}|$, *i.e.* when the observations are more uncertain than the model state. The data assimilated here are relatively accurate, so that $|\boldsymbol{W}| / |\boldsymbol{R}| \cong O(10^{-1})$.

When $\mathcal{L}(\boldsymbol{x})$ is a linear functional and Ξ is an identity mapping, (6) simplifies to

$$\boldsymbol{y}_i = \boldsymbol{x}_i^f - \langle \boldsymbol{x} \rangle^f, \qquad (7a)$$

$$\boldsymbol{\mathcal{L}}(\boldsymbol{y}_i) = \boldsymbol{H}\,\boldsymbol{y}_i, \qquad (7b)$$

$$\boldsymbol{L}\boldsymbol{L}^T = \boldsymbol{H}\boldsymbol{P}^f\,\boldsymbol{H}^T, \quad \boldsymbol{Y}\boldsymbol{L}^T = \boldsymbol{P}^f\,\boldsymbol{H}^T,$$

$$\left[\boldsymbol{H}\boldsymbol{P}^f\,\boldsymbol{H}^T + \boldsymbol{W}\right]\boldsymbol{b}_i = \boldsymbol{d} - \boldsymbol{H}\,\boldsymbol{x}_i^f + \boldsymbol{e}_i, \qquad (7c)$$

$$\boldsymbol{x}_i^a = \boldsymbol{x}_i^f + \boldsymbol{P}^f\,\boldsymbol{H}^T\boldsymbol{b}_i, \qquad (7d)$$

which amounts to applying the usual Kalman filter analysis equations to update each ensemble member in turn.

When the Hadamard products with the compactly supported correlation function are introduced and when the subscript ranges are explicitly written down, (6c) and (6d) are replaced by

$$\text{for}(1 \le p \le n_d,\ \ 1 \le q \le n_d):\ \ c_{pq} = c_{qp} = C_h(r_h^{(pq)})\,C_v(r_v^{(pq)}), \qquad (8a)$$

$$\text{for}(1 \le i \le m):\ \ \left[\boldsymbol{C} \bullet \boldsymbol{L}\boldsymbol{L}^T + \boldsymbol{W}\right]\boldsymbol{b}_i = \boldsymbol{d} - \boldsymbol{\mathcal{L}}(\boldsymbol{y}_i + \langle \boldsymbol{x} \rangle^f) + \boldsymbol{e}_i, \qquad (8b)$$

$$\text{for}(1 \le k \le n_{box},\ \ 1 \le p \le n_d):\ \ \eta_{\boldsymbol{k}\,p} = C_h(r_h^{(kp)})\,C_v(r_v^{(kp)}), \qquad (8c)$$

$$\text{for}(1 \le i \le m,\ \ 1 \le k \le n_{box}):\ \begin{cases} \boldsymbol{\gamma}_{ik} = \boldsymbol{L}^T\boldsymbol{b}_i \bullet \boldsymbol{\eta}_{\boldsymbol{k}}, & (8d) \\ x_{ik}^a = x_{ik}^f + \boldsymbol{y}_{\boldsymbol{k}} \circ \boldsymbol{\gamma}_{ik}, & (8e) \end{cases}$$

where $\circ$ refers to the inner product of two vectors and $\boldsymbol{C}$ ($n_d \times n_d$) is a compactly supported correlation matrix whose elements are defined by (8a), where the indices p and q refer to the data w_p and w_q. The components of the $n_d \times 1$ vector $\boldsymbol{\eta}_{\boldsymbol{k}}$ defined by (8c) contain idealized correlations between the (λ, θ, z) coordinates, of grid box k and the coordinates of each measurement. Note that to simplify the notation only one subscript is used to identify the grid box. The index, $1 \le k \le n_{box}$, thus loops over the three dimensions of the (λ, θ, z) coordinate system. The $m \times 1$ vector, $\boldsymbol{y}_{\boldsymbol{k}} = \{y_{1k}, \cdots, y_{mk}\}$, contains smoothed deviations from the ensemble mean of the m ensemble state vectors in the kth grid box. It is thus a single row of matrix $\boldsymbol{Y}$. In the MvEnKF implementation discussed herein, y_{ik} actually has four components, *i.e.*,

$$y_{ik} = \Xi(\{T, S, u, v\}_{ik} - \{\langle T \rangle, \langle S \rangle, \langle u \rangle, \langle v \rangle\}_k).$$

The $m \times 1$ vector, $\boldsymbol{\gamma}_{\boldsymbol{ik}}$, contains the weights with which the elements of $\boldsymbol{y}_{\boldsymbol{k}}$ are combined in the kth grid box to update the ith ensemble member.

With the horizontal domain decomposition discussed in Sections 3.1 and 3.5, the equations solved on each PE during the analysis become

$$\text{for}(1\le i\le m):\quad \begin{cases} \boldsymbol{y}_i^{\mathcal{C}} = \Xi^{\mathcal{C}}(\boldsymbol{x}_i^f - \langle \boldsymbol{x}\rangle^f), & (9a)\\ \boldsymbol{L}_i^{\mathcal{A}} = \boldsymbol{L}^{\mathcal{A}}(\boldsymbol{y}_i + \langle \boldsymbol{x}\rangle^f) - \boldsymbol{L}^{\mathcal{A}}(\langle \boldsymbol{x}\rangle^f), & (9b)\end{cases}$$

$$\text{for}(1\le p\le n_d^{\mathcal{A}},\ 1\le q\le n_d^{\mathcal{A}}):\quad c_{pq}^{\mathcal{A}} = C_h(r_h^{(pq)})\,C_v(r_v^{(pq)}), \qquad (9c)$$

$$\text{for}(1\le i\le m):\quad \left[\boldsymbol{C}^{\mathcal{A}}\bullet\boldsymbol{L}^{\mathcal{A}}\left(\boldsymbol{L}^{\mathcal{A}}\right)^T + \boldsymbol{W}^{\mathcal{A}}\right]\boldsymbol{b}_i = \boldsymbol{d}^{\mathcal{A}} - \boldsymbol{L}^{\mathcal{A}}(\boldsymbol{y}_i + \langle \boldsymbol{x}\rangle^f) + \boldsymbol{e}_i^{\mathcal{A}}, \qquad (9d)$$

$$\text{for}(1\le p\le n_d^{\mathcal{A}},\ 1\le k\le n_{box}^{\mathcal{C}}):\quad \eta_{kp} = C_h(r_h^{(kp)})\,C_v(r_v^{(kp)}), \qquad (9e)$$

$$\text{for}(1\le i\le m,\ 1\le k\le n_{box}^{\mathcal{C}}):\quad \begin{cases} \gamma_{ik} = \left(\boldsymbol{L}^{\mathcal{A}}\right)^T \boldsymbol{b}_i \bullet \boldsymbol{\eta}_k, & (9f)\\ x_{ik}^a = x_{ik}^f + \boldsymbol{y}_k \circ \boldsymbol{\gamma}_{ik}. & (9g)\end{cases}$$

In (9), $\boldsymbol{y}_i^{\mathcal{C}}$ identifies the part of $\boldsymbol{y}_i$ that corresponds to all grid boxes on the current PE including those contained in the halo regions (area $\mathcal{C}$ in Figure 4). Likewise, $\boldsymbol{d}^{\mathcal{A}}$ denotes the measurements contained within the $\mathcal{A}$ region. The local matrices, $\boldsymbol{C}^{\mathcal{A}}$, $\boldsymbol{L}^{\mathcal{A}}$ and $\boldsymbol{W}^{\mathcal{A}}$, correspond to the global $\boldsymbol{C}$, $\boldsymbol{L}$ and $\boldsymbol{W}$ but only account for the observations contained in $\boldsymbol{d}^{\mathcal{A}}$. As in (9a), the indices *p* and *q* in (9c) refer to the *p*th and *q*th observations. The measurement functional, $\boldsymbol{L}^{\mathcal{A}}(\boldsymbol{x})$, maps the global state vector, distributed across PEs, to the elements of $\boldsymbol{d}^{\mathcal{A}}$. With the form of $\boldsymbol{L}^{\mathcal{A}}$ considered herein, the mapping does not necessitate an exchange of information between PEs (Section 3.9). The smoothing function, $\Xi^{\mathcal{C}}$, returns the local elements of the global vector returned by Ξ in (6a).

To update the state variables of the *i*th ensemble member in grid box *k*, $\{T, S, u, v\}_{ik}$, the analysis update, (9e-g) or (8c-e), involves *m* matrix-vector multiplication of $\boldsymbol{L}^T$ by $\boldsymbol{b}_i \bullet \boldsymbol{\eta}_k$ (8d, 9f). If the state variables were not distributed across PEs, or if the observations allowed to influence the variables of each grid box were not limited to a sub-region of the entire domain as a result of imposing compactly supported background covariances, these multiplications would be costly. Yet, for the Poseidon model distributed across 256 PEs, the number of matrix-vector products on each PE drops to $m(i_2 - i_1 + 1)(j_2 - j_1 + 1)K \approx 32000$, where a typical size of $\boldsymbol{L}^T$ is 40×100. Although these products take up most of the time spent in the analyses, they correspond to a tolerable fraction of the total cost of the MvEnKF—most of which is associated with the error-covariance forecast (ensemble integration).

The above remark illustrates how the paradigm shift from serial algorithms to massive parallelism ones enables one to conveniently solve problems that can not be addressed in a traditional scientific computing environment consisting in vector supercomputers. Yet, the massively parallel solution of a complex numerical problem usually requires *ad hoc* approximations. The crucial approximation made here is the confinement of the analysis (Section 3.5) that results from relying on a compactly supported covariance model (Section 3.4). The parallel distribution of the calculations naturally follows. It is shown in Section 4.1 that the

impact of this approximation on the assimilation increments is negligible.

3.7 Incremental analysis

Incremental analysis updating (IAU, *e.g.*, Bloom *et al.* 1995) is used to insert the analysis increments, $\boldsymbol{x}^a - \boldsymbol{x}^f$, into the model in a gradual manner. Namely, the model partial differential equations (1-4) are replaced with

$$\frac{\partial \boldsymbol{x}}{\partial t} = \boldsymbol{F}(\boldsymbol{x}, t) + \frac{(\boldsymbol{x}^a(t_i) - \boldsymbol{x}^f(t_i))}{(t_{i+1} - t_i)}, \quad t_i \le t < t_{i+1}, \qquad (10)$$

where $\boldsymbol{F}$ stands for the right hand sides of (1-4) and $\boldsymbol{x}^a(t_i)$ and $\boldsymbol{x}^f(t_i)$ are the analysis and forecast at the time, t_i, of the ith analysis.

Unlike nudging (*e.g.*, Daley 1991), which relaxes the model state toward an analysis, the analysis increments are inserted as a state-independent forcing term. The IAU has properties similar to those of a low-pass filter and can improve observed-minus-forecast statistics with respect to a non-incremental updating scheme (Bloom *et al.* 1995).

The IAU is used here for two reasons. First, it lessens the unwanted effects of intermittent data assimilation, specifically initialization shocks resulting from imbalances between the model fields following the direct insertion of the analysis increments. Second, the IAU allows the model to gradually adjust the *h* field in response to the *T*, *S*, *u* and *v* increments without violating the constraints imposed by the continuity equation (1).

3.8 Measurement functional

The data processed in oceanographic data assimilation are usually current, temperature or salinity measurements made inside the model domain. Alternatively, the data sometimes measure an integrated quantity such as sea surface height, heat content or, in acoustic tomography, travel time.

In the application discussed here, the measurement functional, $\mathcal{L}^a(\boldsymbol{x})$, is simply a 2D interpolation operator which maps the model temperature field—previously interpolated vertically to a set of levels which include the depths of the measurements—to the latitude and longitude of each observation on the appropriate depth level.

Prior to the interpolation, bisection (*e.g.*, Knuth 1998) is used to find the grid box, (*i, j, k*), containing each measurement. Then, quadratic polynomials passing through the grid cell containing each measurement and the eight grid cells which border that cell are used to complete the mapping. Four polynomials are used per observation (Fig. 5b). Unlike biquadratic splines, this type of polynomial fitting does not guarantee the continuity of the first derivative across grid cell boundaries. It is however less costly than spline fitting when the observations are scarce or far apart, as is common in oceanography.

Each PE performs the interpolation to the locations of the observations, $\boldsymbol{d}^{\boldsymbol{b}}$, contained within its PE-private area (Fig. 5a). Due to the presence of the halo regions, the horizontal interpolation can be made without exchanging information between neighboring PEs.

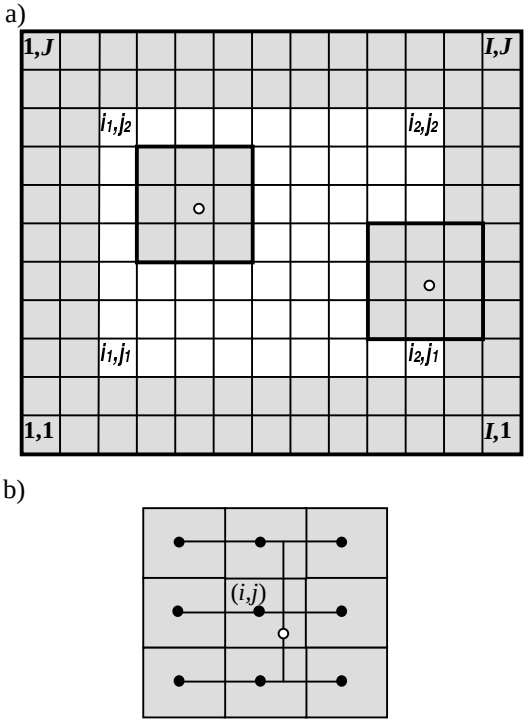

Figure 5. Illustration of horizontal interpolation in measurement functional. (a) Grid cells involved in interpolation called for by $\boldsymbol{L}^{\boldsymbol{a}}(\boldsymbol{x})$. Two cases are illustrated. The white circles indicate the position of the observations located inside the unshaded PE-private area. The gray-colored rectangles show which grid cells contain the state-vector elements needed to complete the interpolation. The presence of the halo regions (outer gray areas) saves the communication cost associated with requesting information from nearby PEs when the measurements are located near the boundary of the PE-private area. (b) Horizontal interpolation mechanism using four quadratic polynomials for each observation represented by the white circle in grid cell (*i*, *j*).

3.9 Superobsevations

As is common when several measurements are made at the same location between successive analyses, the observations are smoothed temporally. This operation, sometimes referred to as superobing and introduced by Lorenc (1981), combines the measurements using weights which decrease exponentially with the time interval between the time, $t_i + \delta$, of a measurement and that, t_i, of the *i*th analysis:

$$d(t_i) = \tfrac{1}{2\Theta}(1-e^{-\frac{\Delta}{\Theta}})^{-1}\int_{-\Delta}^{+\Delta} w(t_i+\delta)e^{-\frac{|\delta|}{\Theta}}d\delta. \qquad (11)$$

In (11), Θ is a time scale, $d(t_i)$ is a superobservation and the w are individual measurements made between time $t_i - \Delta$ and time $t_i + \Delta$. Here and in KR01, $\Theta = \Delta = 10$ days and the analyses occur every five days (TAO temperature data) or daily (TOPEX altimeter data).

3.10 Pre-filtering

The purpose of the smoothing operator, Ξ in (6-9), is to remove spurious short-range covariances from the representer matrix, $\boldsymbol{R}$. These spurious elements result from the limited ensemble size used to estimate the error distribution and from associated sampling errors. Spurious long-range covariances are filtered out by imposing that the covariance functions be compactly supported (Section 3.4).

The operator Ξ relies on a simple one-dimensional recursive (infinite impulse response) filter which is applied horizontally in each layer to damp small-scale variability prior to calculating $\boldsymbol{L}$. The filter equations are

$$y_i^a = \frac{1}{1+\omega_c}\left[\omega_c\left(x_{i-2} - x_i\right) + 2y_{i-1}^a - (1-\omega_c)\,y_{i-2}^a\right], \qquad (12a)$$

$$y_i^b = \frac{1}{1+\omega_c}\left[\omega_c\left(x_{i+2} - x_i\right) + 2y_{i+1}^b - (1-\omega_c)\,y_{i+2}^b\right], \qquad (12b)$$

$$y_i = \frac{1}{n}\left[(i-1)y_i^a + (n-i+1)y_i^b\right], \qquad (12c)$$

where $\omega_c = \tan(f_c)$ and $0 \le f_c \le \frac{\pi}{2}$ is the cutoff frequency. The filter input and output are x_i and y_i, $1 \le i \le n$. Note that the term recursion denotes a spatial recursion: i is a subscript into vector $\boldsymbol{y}$.

Unlike non-recursive filters (*e.g.*, a Shapiro filter or a running mean), which have polynomial response functions, recursive filters have rational response functions which make it easier to design a filter with a sharp response (*e.g.*, Hamming 1983).

Three passes of the filter are used with $f_c = \frac{\pi}{4}$. Each pass consists of applying (12) successively along the zonal and meridional directions. Prior to the filter's application, the ensemble mean is subtracted from each ensemble member's state vector, as indicated in (6a) and (9a).

Before applying (12), each PE collects the state-vector elements from the PEs which belong to the same row (zonal application of the filter) or column (meridional application of the filter) of the PE lattice (Fig. 1). For each input array $\boldsymbol{x}$ in (12), every PE calculates the entire array $\boldsymbol{y}$ and discards the array elements it does not need. Because of the latency time associated with each message, this approach is less costly than letting each PE wait until the downstream PE has finished computing its own elements of $\boldsymbol{y}$ and has sent the corresponding end values to the current PE before starting the PE's own calculations, and so forth. To circumvent memory limitations, the filtering is applied to the *T*, *s*, *u* and *v* fields of each ensemble member in turn.

3.11 System-noise representation

The theory of the Kalman filter (*e.g.*, Gelb, 1974) assumes that the first- and second-order statistics of the unknown errors in the model and external forcing are known. Higher-order statistics are neglected. Let the evolution of the true state be represented by

$$\frac{\partial \boldsymbol{x}}{\partial t} = \boldsymbol{F}(\boldsymbol{x},t) + \boldsymbol{\xi}(\boldsymbol{x},t), \qquad (13)$$

where $\boldsymbol{\xi}$ combines the model errors and forcing errors, and is commonly known as system noise or process noise. As in (11), $\boldsymbol{F}$ is the vector of right hand sides of (1-4) which includes the model hydrodynamics, physics and forcing. It is assumed that the model and forcing are unbiased, *i.e.* $\langle \boldsymbol{\xi}(\boldsymbol{x},t) \rangle = 0$, and that the $\boldsymbol{\xi}$ vectors are uncorrelated in time:

$$\langle \boldsymbol{\xi}(\boldsymbol{x}_k,t_k)\,\boldsymbol{\xi}(\boldsymbol{x}_l,t_l) \rangle = \boldsymbol{\Gamma}(\boldsymbol{x}_k,\boldsymbol{x}_l)\delta(t_k - t_l), \qquad (14)$$

where the system-noise covariance matrix, $\boldsymbol{\Gamma}$, is assumed known. Of course, the unbiased assumption is rarely correct in practice. This is especially true with ocean models in which the thermocline layer is usually too diffuse. The latest version of the assimilation code includes an algorithm, derived from Dee and Da Silva (1998), to estimate and correct systematic model errors. The issue of correcting the model bias with the EnKF will be discussed in a separate paper.

In meteorological and oceanographic data assimilation, the statistics of $\boldsymbol{\xi}$ are generally unknown and are the object of parameterization. Adaptive Kalman filters that simultaneously estimate the state and system-noise statistics have been developed. However, the prohibitive cost of the adaptive filters has limited their application. Blanchet and Frankignoul (1997) summarize and compare several adaptive filtering algorithms.

Motivated by the current lack of information about the model-error statistics, the system-noise is represented solely by modeling the errors in the surface wind stress and heat flux forcing. A system-noise representation in which not only the forcing errors but also the model errors are parameterized is in development.

Because of the focus on seasonal-to-interannual variability, the forcing errors (uncertainties) are modeled on those time scales, with each ensemble member being forced by a monthly mean perturbation of the monthly mean basic state. The basic state is the superposition of the climatological seasonal cycle with interannual anomalies. The climatology is provided by Special Sensor Microwave Imager (SSM/I: Atlas *et al.* 1996) winds and Earth Radiation Budget Experiment (ERBE) heat flux data. The interannual anomalies are provided by the atmospheric model integrated over observed SST data (Reynolds and Smith 1994). The perturbations applied are due entirely to internal atmospheric chaos and are generated by starting the atmospheric integration at different times. By using the same SST, each member of the atmospheric ensemble used to force the ocean ensemble has the same seasonal and interannual phase. The

spread of the atmospheric ensemble from which the forcing anomalies are derived is meant to be representative of the uncertainty of the forcing products used to force the model in non-ensemble runs.

3.12 Inter-processor communications

All information exchanges between PEs, during the analysis as well as during the error-covariance forecast (ensemble integration), use message-passing functions from the Goddard Earth Modeling System (GEMS, Schaffer and Suarez 1998) library. The GEMS functions provide a high-level, object oriented interface to the CRAY native SHMEM (shared memory) communication library.

The position of each PE on the lattice is stored in the $n_{PE} \times m_{PE}$ array ***PE***, where n_{PE} and m_{PE} are the number of PEs along the zonal and meridional directions, respectively. In this implementation, $n_{PE} = m_{PE} = 16$. The total number of PEs is N_{PE}. Every PE has a copy of ***PE***. Some tasks, such as accessing external files, are always done by the same PE which is referred to as **`root`**.

The assimilation algorithm relies mostly on two GEMS functions to exchange information between PEs. These two functions are mentioned here in template form to simplify the discussion of Section 3.13. The first function, **`pe_collect(...)`**, is used to collect data from either the entire ***PE*** array or from the row or column of ***PE*** which contains the current PE. The second function **`halo(...)`** updates its array argument in the halo regions of each PE after the PEs have modified the PE-private part of this array.

3.13 Parallel algorithm

The assimilation algorithm, various aspects of which were discussed in the preceding Sections, contains the following steps which are listed from the point of view of one PE, hereafter referred to as the current PE. The enumeration of these steps starts after the current PE has obtained the observations, $\boldsymbol{d}^b$, made within its PE-private region ($\boldsymbol{\mathcal{B}}$ in Fig. 4) from **`root`**. The task of reading all the data and broadcasting them is assigned to **`root`**. The current PE then extracts the data that fall into its PE-private area. In the array ***PE***, The current PE is $PE_{ic\,jc}$.

- Step 1: Vertical interpolation of the *T*, *S*, *u* and *v* fields from the isopycnal model layers to the analysis levels as explained in Section 3.2.
- Step 2: Calculation of the anomalies with respect to the ensemble mean over the entire domain of the current PE, $\boldsymbol{x}_i^{cf} - \left\langle \boldsymbol{x}^{c} \right\rangle^{f}, 1 \le i \le m.$
- Step 3: Calculation of $\boldsymbol{y}_i^{c}$, as in (9a). Prior to each zonal application of the filter (12), a call to **`pe_collect( )`** is used to collect the state elements required to run the recursive filter from the PEs listed in column *jc* of ***PE***. The same holds for each meridional application of (12), where row *ic* of ***PE*** is now involved.

- Step 4: Identification of the PE-private data required by the other PEs. First, **pe_collect( )** is used to collect the longitudes and latitudes of each PE's southwestern, southeastern, northwestern and northeastern corner grid cells. Using this information, the current PE calculates for each (*i*, *j*) pair which elements of $\boldsymbol{d}^{\boldsymbol{b}}$ fall inside the rectangle, $\mathcal{A}_{ij}$, which is the region from which PE_{ij} will need to collect data (Fig. 4). The indices of the relevant elements of $\boldsymbol{d}^{\boldsymbol{b}}$ are stored in the array $\boldsymbol{k}_{ij}$.

- Step 5: Evaluation of the measurement functional. The current PE calculates a $n_d^{\boldsymbol{b}} \times m$ matrix, $\boldsymbol{L}^{\boldsymbol{b}}$ where $n_d^{\boldsymbol{b}}$ is the number of observations contained in its own PE-private region. The element at the intersection of the *p*th row and *i*th column of $\boldsymbol{L}^{\boldsymbol{b}}$ is

$$L_{pi}^{\boldsymbol{b}} = \mathcal{L}^p(\boldsymbol{y}_i + \langle \boldsymbol{x} \rangle^f) - \mathcal{L}^p(\langle \boldsymbol{x} \rangle^f),$$

  where $\mathcal{L}^p$ is the interpolation operator which maps its argument to the location of $d_p^{\boldsymbol{b}}$, the *p*th PE-private observation on the current PE (Section 3.8).

- Step 6: Calculation of $\boldsymbol{z}^{\boldsymbol{b}}$, the innovations with respect to the ensemble mean for the current PE's private region. The innovation corresponding to $d_p^{\boldsymbol{b}}$ is

$$z_p^{\boldsymbol{b}} = d_p^{\boldsymbol{b}} - \mathcal{L}^p(\langle \boldsymbol{x}^c \rangle^f).$$

- Step 7: Gathering of $\boldsymbol{L}^{\boldsymbol{a}}$ on each PE using the information recorded in the $\boldsymbol{k}_{ij}$ arrays. The function **pe_collect( )** is called N_{PE} times. Each call results in a different PE completing the collection of its version of $\boldsymbol{L}^{\boldsymbol{a}}$.

- Step 8: Collection of the innovations, $\boldsymbol{z}^{\boldsymbol{a}}$, required by each PE. As for gathering $\boldsymbol{L}^{\boldsymbol{a}}$, **pe_collect( )** is called N_{PE} times. Each PE passes to **pe_collect( )** the elements of its $\boldsymbol{z}^{\boldsymbol{b}}$ innovation vector required by the other PEs. The PEs now have all the information required to calculate the analysis increments.

- Step 9: Calculation of the representer amplitudes. First the local representer matrix, $\boldsymbol{R}^{\boldsymbol{a}} = \boldsymbol{L}^{\boldsymbol{a}}(\boldsymbol{L}^{\boldsymbol{a}})^T$, and its Hadamard product with the compactly supported correlation function, $\mathcal{C}^{\boldsymbol{a}} \bullet \boldsymbol{R}^{\boldsymbol{a}}$, are computed. Then the *m* right hand sides of (9d) are calculated as $\boldsymbol{z}^{\boldsymbol{a}} - \boldsymbol{L}_i + \boldsymbol{e}_i, 1 \le i \le m$, where $\boldsymbol{e}_i$ is the random perturbation term of (6c). Finally, (9d) is solved *m* times, yielding the $\boldsymbol{b}_i$ vectors. Since the effective rank of $\boldsymbol{R}^{\boldsymbol{a}}$ is *m* rather than $n_d^{\boldsymbol{a}}$ and as a precaution against $\boldsymbol{R}^{\boldsymbol{a}}$ losing its positive definiteness due to round off errors, LU decomposition with partial pivoting is used rather than Cholesky decomposition. If LU decomposition fails, singular value decomposition (SVD) is used and near-zero singular values of $\boldsymbol{S}^{\boldsymbol{a}}$ are ignored.

- Step 10: Computation of the analysis increments. The calculations (9e-g) are made for each PE-private grid box. Calls to **`halo( )`** are used to fill the elements of $\boldsymbol{x}^a - \boldsymbol{x}^f$ in the current PE's halo regions. It is more economical to obtain these elements in this manner than through the application of (9e-g) to each grid box situated within the halo regions.
- Step 11: Transformation of the *T*, *S*, *u* and *v* increments from the analysis levels to averages on the model layers. This step is the reciprocal of step 1. Following this, the analysis increments are added gradually to each ensemble member's state vector by means of the IAU mechanism discussed in Section 3.7.

4 Discussion

4.1 Effect of parallel decomposition on analysis

The impact of performing a different local inversion on each processor (9d) rather than inverting the global system matrix, $\boldsymbol{S}=\boldsymbol{C}\bullet\boldsymbol{R}+\boldsymbol{W}$ in (8b), is examined in this Section. So that the local and global solutions can be compared, a single TAO temperature analysis is used as an example because it involves sufficiently few data for (8b) to be solved on each PE without partitioning $\boldsymbol{S}$ as in (9d).

The parallel algorithm relies on the assumption that (1) the analysis calculations can be partitioned resulting in each processor assimilating local data and that (2) the partitioning does not have a deleterious effect on the analysis results. An alternative approach when presented with many data to assimilate simultaneously is to solve the global problem (6c) with an iterative method. The NASA Data Assimilation Office's Physical Space Statistical Analysis System (Cohn *et al.* 1998) and the Naval Research Laboratory Variational Data Assimilation System (Daley and Barker 2001) use a preconditioned conjugate gradient solver (PCGS) to solve a system akin to (6c). A similar algorithm has been implemented into the NSIPP multivariate data assimilation system (MvDAS). This iterative solver is faster than LU decomposition for $n_d \geq O(10^3)$. So far, the number, n_d^a, of data processed on each PE have been less than that. Thus, LU decomposition or SVD has been used most of the time (Section 3.13).

Here, for illustrative purposes, a 25-member ensemble distributed across 100 PEs is used. The experiments of KR01 involve a 40-member ensemble and 256 PEs. There are 642 temperature measurements in this example. They correspond to January 1, 1997. Although the number of observations does not necessitate distributing the analysis computations, the example illustrates how the inversion would be distributed if there were too many data for each PE to process them all at one time, as is the case when TOPEX altimeter data are assimilated.

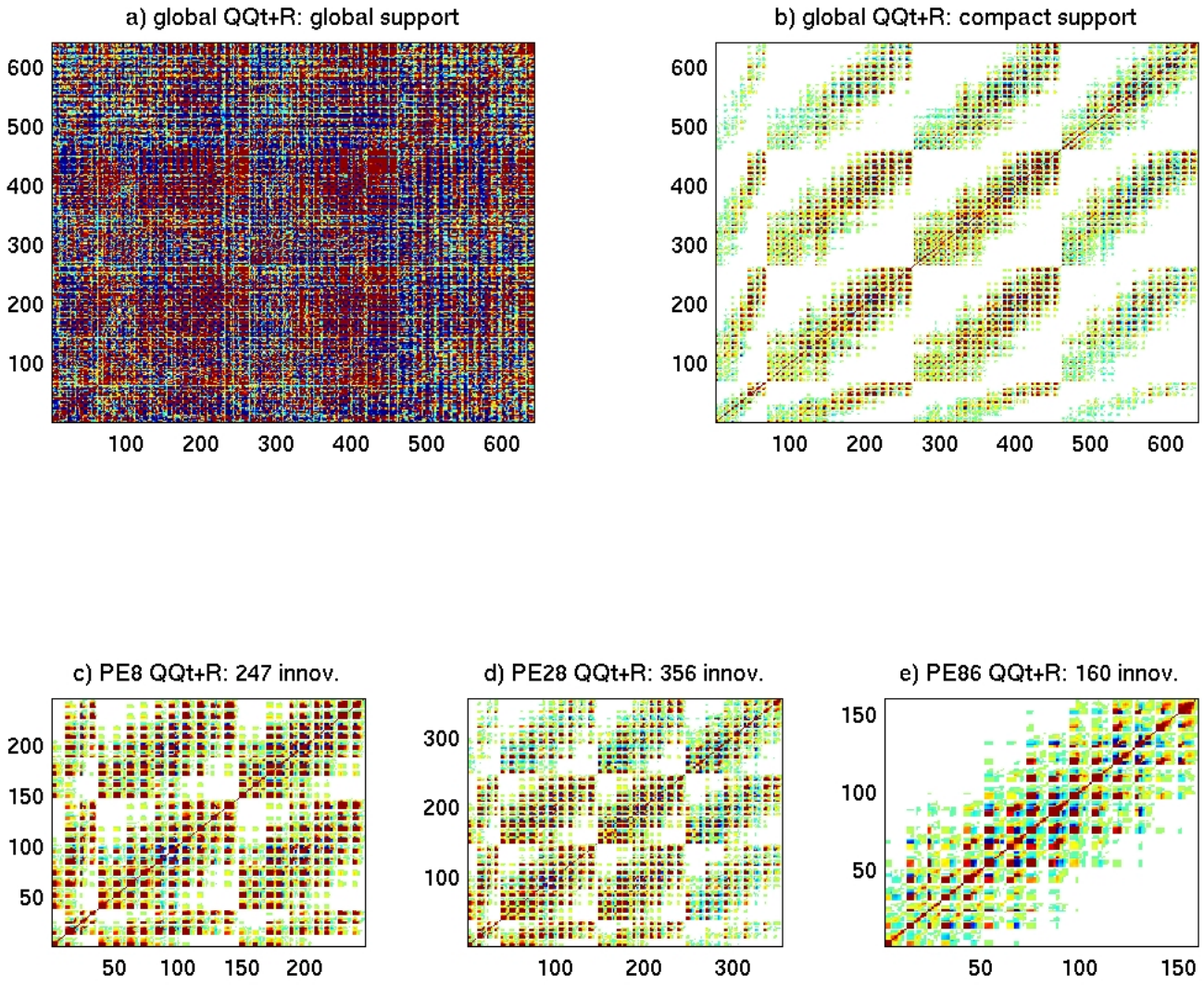

Figure 6. Structure of error-covariance matrices in observation space for one TAO temperature analysis corresponding to January 1, 1997. (a) Global system matrix, ***S***, without compact support. (b) Global compactly supported ***S***. (c-e) Example PE-local ***S*** matrices corresponding to PE 8 for which $n_d = 247$, PE 28 ($n_d = 356$) and PE 86 ($n_d = 160$).

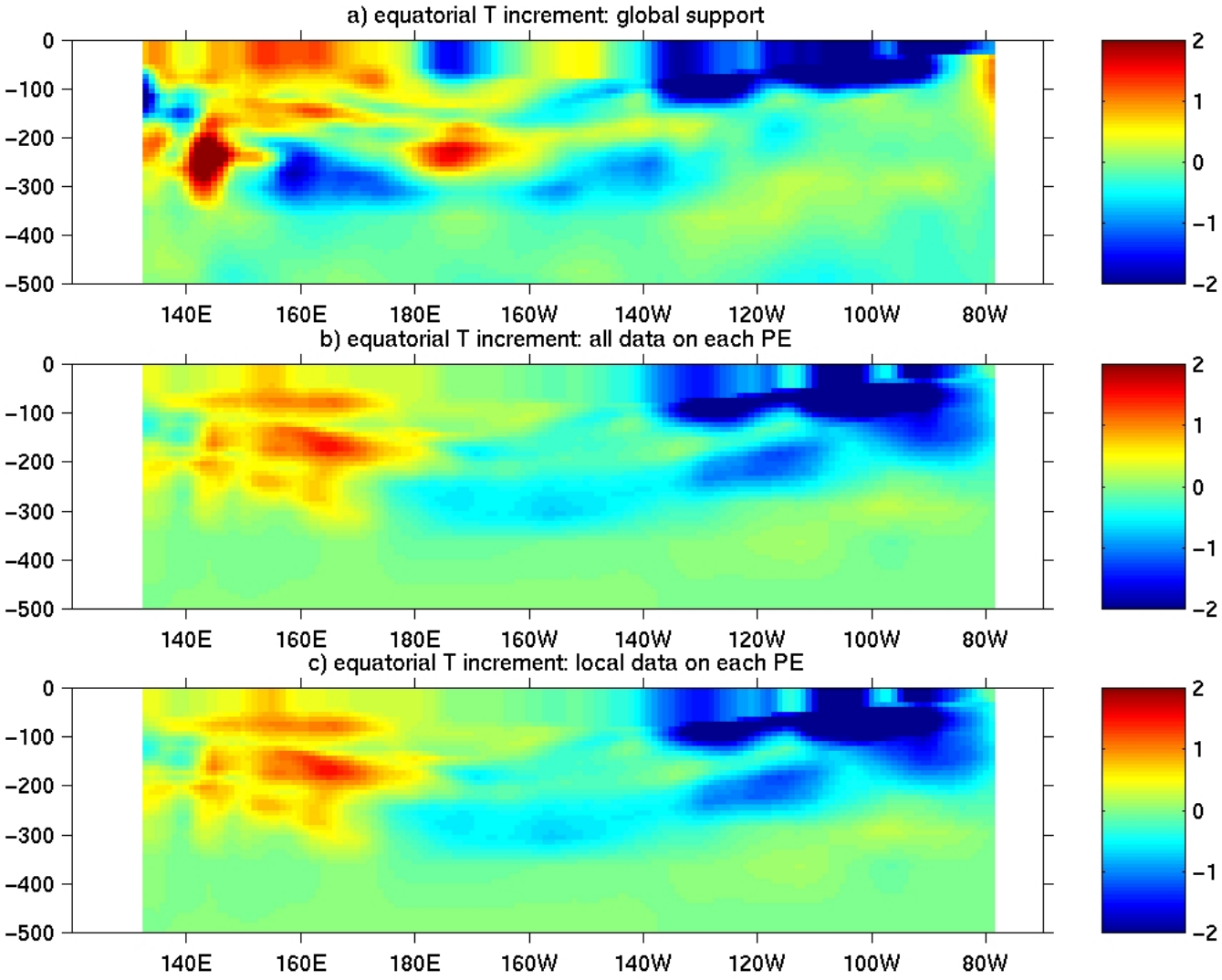

Figure 7. Equatorial sections through the temperature-field part of the analysis increments (degrees C) corresponding to the cases shown in Fig.6. (a) Global inversion without compactly supported covariances. (b) Global inversion with compact support. (c) Distributed inversion with compact support.

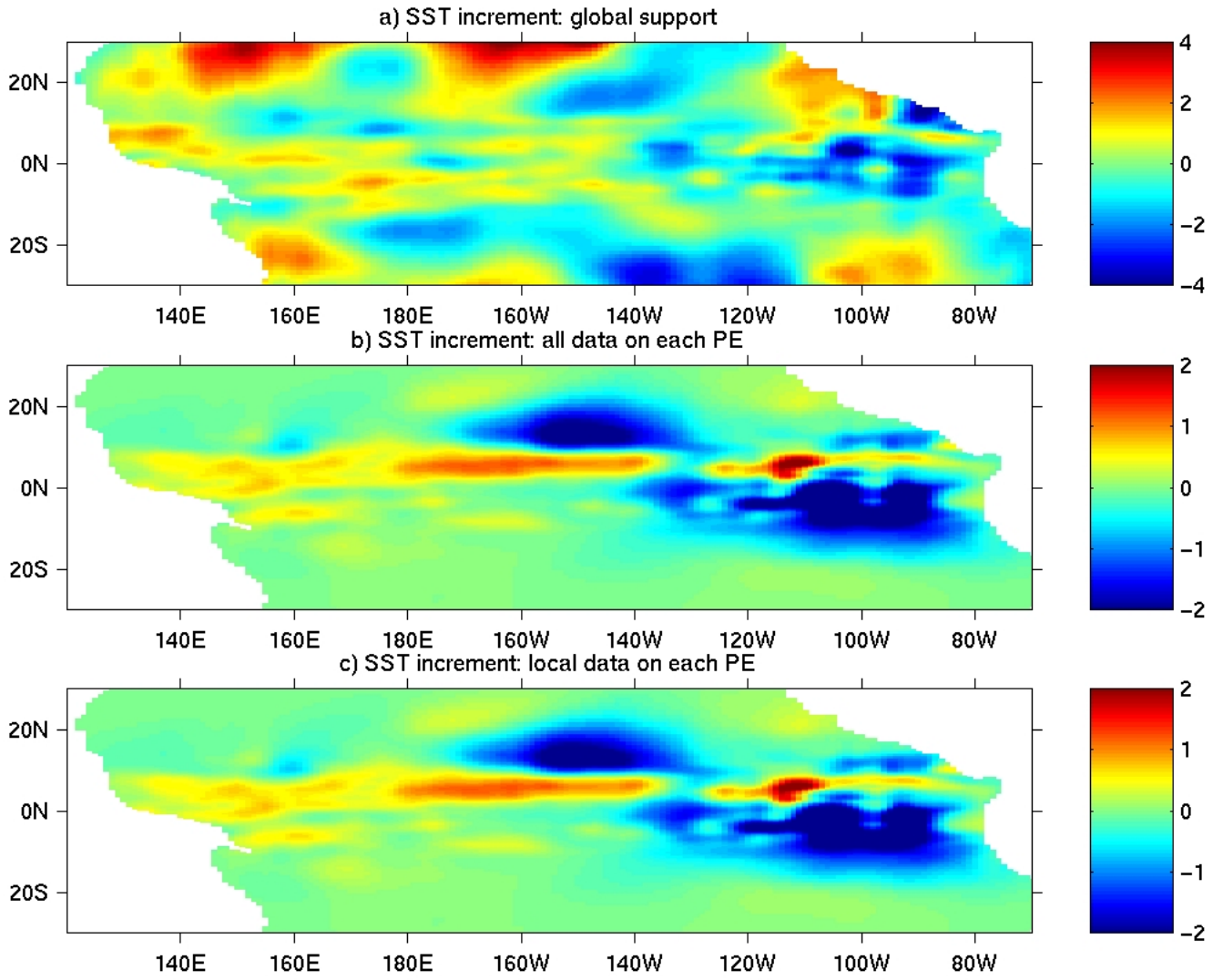

Figure 8. Same as Fig. 7 for the sea-surface temperature increments.

Figure 6 shows how imposing compact support to $\boldsymbol{R}$ impacts the sparseness of the global $\boldsymbol{S}$. It also illustrates how the sparseness is exploited by distributing the analysis calculations in the parallel algorithm. Figures 7 and 8 illustrate the respective impacts on the assimilation increments of using compactly supported background covariances and distributing the analysis among PEs. As is common, a diagonal $\boldsymbol{W}$ is assumed.

Figure 6a shows the global $\boldsymbol{S}$, when the condition that it be compactly supported is not imposed ($\boldsymbol{L}\boldsymbol{L}^T+\boldsymbol{W}$ in 6c). Figure 7a shows an equatorial section through the corresponding temperature increment. The corresponding sea-surface temperature (SST) increment is shown in Figure 8a.

When the background covariances are compactly supported, the global $\boldsymbol{S}$ ($\boldsymbol{C}\bullet\boldsymbol{L}\boldsymbol{L}^T+\boldsymbol{W}$ in 8b), becomes sparse as Figure 6b illustrates. The most obvious effect of the Hadamard product of $\boldsymbol{C}$ and $\boldsymbol{R}$ on the assimilation increment is that the latter is tapered away from the Equator where no measurements are available (Fig. 8b). The effect of the Hadamard product on the vertical structure of the temperature increment is not as dramatic (Fig. 7b) since the data come from several depths between the surface and 500 meters.

When the analysis is distributed, the calculation of the local $\boldsymbol{S}$ on each PE ($\boldsymbol{C}^a\bullet\boldsymbol{L}^a\left(\boldsymbol{L}^a\right)^T+\boldsymbol{W}^a$ in 9d) amounts to sub-sampling the global compactly supported $\boldsymbol{S}$ of Figure 6b. On each PE, the sub-sampling results in a local $\boldsymbol{S}$ which is less sparse than the global $\boldsymbol{S}$ because it does not contain covariances between remote locations which are identically zero as a result of the Hadamard product. Figures 6c-e show local $\boldsymbol{S}$ matrices on three randomly chosen PEs.

Comparing Figure 7c to Figure 7b or Figure 8c to Figure 8b shows that the analysis increments obtained with the local analysis equations (9) are virtually identical to those obtained with (8), even though the global inversion (8b) is bypassed. Indeed, the root mean square difference between the Equatorial temperature increments of Figures 7b and 7c is 6.0×10^{-4} C. That between the SST increments of Figures 8b and 8c is 1.0×10^{-3} C. Thus, the tremendous computational savings associated with substituting the local $\boldsymbol{S}$ for the global $\boldsymbol{S}$ occur with a negligible impact on the quality of the analysis.

4.2 Timing

Table 1 lists the wall-clock time spent in each step of the assimilation algorithm and in the ensemble integration in the case of a one-month TAO temperature assimilation experiment (TAOA: middle column) and in that of a one-month assimilation experiment with gridded TOPEX-altimeter data (TOPA: right column). The operation labeled "data processing and distribution" refers to the root PE reading the observations and broadcasting them to the other PEs. Each PE then determines which data fall within its PE-private region and superobs them (Section 3.9). The remaining, non PE-private data are discarded.

The TAOA example involves approximately 600 data per analysis. The TOPA example involves about 10^4 data per analysis. Although the TOPEX data are processed daily in KR01, they are assimilated every five days in the TOPA run to facilitate comparison to the TAOA run which is made with a five-day assimilation interval. The times listed correspond to one five-day cycle and

are averages over the lengths of the experiments. As in KR01, 40 ensemble members distributed across 256 PEs are used.

In both TAOA and TOPA, a little more than 1000 seconds are spent time stepping the ensemble (error-covariance forecast). The four-second difference between the two cases results from differences in the time spent in disk access, which varies with the system load, and, to a lesser extent, in the time spent communicating between PEs and synchronizing the computations. In TAOA, the ensemble integration takes 72% of the time. In TOPA, it takes 62% of the time. The remainder of the time (TAOA: 395 seconds, TOPA: 626 seconds) is used to process the data.

Table 1. Mean duration of each phase of the analysis cycle. Middle column: TAO temperature assimilation (TAOA in text). Right column: TOPEX altimeter data assimilation (TOPA in text).

Operation	**TAOA Wall-clock time (s)**	**TOPA Wall-clock time (s)**
Five-day ensemble integration	1035	1039
Data processing and distribution	25	107
Step 1	5	5
Step 2	4	4
Step 3	19	17
Step 4	3	5
Step 5	12	93
Step 6	3	17
Step 7	14	19
Step 8	5	6
Step 9	4	37
Step 10	292	307
Step 11	9	9
Total	1430	1665

Steps 1-3 and 10 are independent of the number and nature of the data and thus take the same amount of time in TAOA and TOPA. Most of step 4 is spent communicating, and the number and length of the messages involved is the same in TAOA and TOPA. Therefore, the time spent in this step changes little between the two cases.

The differences between the times spent in steps 5 and 6 are due to the number of data processed being larger in TOPA. Yet, the cost of these steps is not directly proportional to n_d because the $\mathcal{L}$ operator is more expensive in TAOA, since more lookups are involved per datum. Indeed, the data assimilated in the TOPA example have been interpolated onto the model grid before runtime. The cost of mapping the model state to the observations is reduced as a result.

Although about 20 times more data are processed in TOPA than in TAOA, the cost of steps 7 and 8 is roughly the same in TAOA and TOPA. The reason is that these steps consist mostly in communications and the number of messages exchanged is the same in both cases. Only their length changes. Step 9, on the other hand, involves the solution of (9d) which scales like $(n_d^a)^3$. In TAOA, $n_d \approx 600$ while $0 < n_d^a < 200$. In TOPA, $n_d \approx 10^4$ and $n_d^a \approx 10^3$. Thus, step 9 is the step with the most data dependent cost.

A large fraction of the analysis time is spent in Step 10 (TAOA: 74%, TOPA: 49%), the calculation of the analysis increments in each PE-private grid box. The reason for this is the matrix-vector products of (9e). When the contribution of C_v is removed from the compactly supported correlation function used in (9d) and (9e), *i.e.* when $C_v(r_v)=1$ for all r_v, the time taken to complete Step 10 drops to 23 seconds in the TAOA example, *i.e.* by more than 90%. The reason is that the $\boldsymbol{\gamma}_{ik}$ vectors become independent of the analysis level. Thus, rather than K matrix-vector products, a single product is needed to calculate the analysis increment within each grid cell (i, j). Nevertheless, the experience accumulated so far with the MvEnKF has shown that the computational savings associated with setting $C_v=1$ occur to the detriment of the filter's skill.

In summary, the time spent in one TOPA analysis is less than twice that spent in one TAOA analysis, although many more data are involved. The significant efficiency gain of TOPA over TAOA is attributable to two factors. First, the l_λ and l_θ correlation scales used in TOPA are half those used in TAOA. Thus, although n_d is 20 times larger in TOPA than in TAOA, TOPA's $n_d^{\boldsymbol{a}}$ is not 20 times more than TAOA's. Second, the load is nearly optimally balanced in TOPA where the data cover the whole domain. To the contrary, most off-equatorial PEs in TAOA are idle during the analysis. The parallel algorithm is thus much more efficient for a large number of evenly distributed data.

In a serial algorithm, the cost of step 9 in TOPA would be overwhelming unless the PCGS algorithm were used to solve (6c) or (8b). Since the PCGS solver iterates using matrix-vector products of $\boldsymbol{S}$ with $\boldsymbol{b}$, it scales like n_d^2 rather than like n_d^3. In the parallel algorithm, a non-iterative $O(n_d^3)$ solver can be used with no significant penalty (Section 4.1).

For reference, when the TAO temperature data are assimilated using the UOI, the time spent in one complete analysis cycle on 64 (*vs.* 256) PEs is 151 seconds, *i.e.* about 11% of the time taken by the MvEnKF in TAOA. Of these, 61 seconds are used to integrate the model for five days, 27 seconds are spent in preprocessing and distributing the data and 63 seconds are taken by the analysis.

4.3 Scaling

This Section discusses the two main current limitations of the parallel MvEnKF: (1) that it scales poorly beyond 100 PEs in the present machine/model configuration and (2) that the maximum ensemble size attainable is dictated by the memory of the individual PEs on a MPP with distributed memory. A CRAY T3E-600 with 128MB RAM per PE is used here. Before examining these two points, it is worth pointing out that they are of little long-term importance. Whether the implementation would be most efficient on the CRAY T3E on which it was developed matters little because this machine will have been phased out before the MvEnKF becomes mature enough to replace the UOI in the coupled global forecasting system. The expected lifetime of a modern supercomputer is about two years. Therefore, a main objective of the flexible, object-oriented message-passing software engineering approach used to implement

the MvEnKF and the other MvDAS components has been that they be adaptable and easily portable to any parallel platform, small-scale parallel or MPP.

In Figure 9a, it is shown how t_m, the time spent per ensemble member in each five-day analysis cycle in the TAOA run (Section 4.2), scales with N_{PE} (diamonds). The dashed curve labeled "EnKF perfect" extrapolates the value of t_m for 16 PEs in the range from 16 to 256 PEs, assuming linear scaling. According to Amdhal's law, such scaling can never be achieved. He predicted that the speedup attainable in a parallel computing environment can not be linear as there should always come a point where further task division creates more overhead than computational speedup. Instead, the time used by an algorithm on p PEs is given by $t_p = t_s(f + (1-f)/p)$, where t_s is the time used by the same algorithm on a serial machine and f is the fraction of the operations that must be performed sequentially.

The observed scaling is hard to compare with theory. First, because t_s is unknown. Second, because f depends on N_{PE}. For example, the ratio of the size of the halo regions to that of the PE-private regions increases with N_{PE}: the former is essentially dictated by the finite differencing scheme while the latter decreases when N_{PE} increases. Also, the scaling numbers shown involve different ensemble sizes for different values of N_{PE} (see Figure 9b). Still, t_m decreases by a mere 16% when N_{PE} doubles from 128 to 256. Rather, t_m decreases by 45% between 16 PEs and 32 PEs. This is indicative of saturation. The horizontal resolution of the Pacific basin version of Poseidon used in these experiments is not high enough for the distribution of its state vector over more than 100 PES to be optimal. In contrast, the global ocean component of the NSIPP coupled model to which the MvEnKF will be applied next has enough state variables to warrant its distribution over more than 100 PEs.. For reference, the observed and perfect scaling curves are also shown for the UOI. In this case, the saturation becomes apparent with 64 PEs at the current model resolution.

In figure 9b, the largest ensemble size allowed by the individual-PE memory on NSIPP's current computational platform, m_{max}, is shown as a function of N_{PE}. For each value of N_{PE}, the timing number in Figure 9a corresponds to m_{max} ensemble members, so that memory is saturated. Between 16 PEs and 128 PEs, m_{max} increases approximately linearly from 6 to 36. On 256 PEs, m_{max} is 46. To increase m_{max} for given N_{PE}, the following approach can be used: partition the ensemble so as, for example, to run 16 6-member sub-ensembles on 16 PEs each, for a total of 96 members on 256 PEs. This is easier said than done because it would require a communication mechanism present in the Message Passing Interface (MPI, Message Passing Interface Forum 1994) but not currently supported by the GEMS library. Alternatively, running the MvEnKF on a platform with globally addressable memory would also allow larger ensemble sizes. The cost of the MvEnKF would obviously be higher in both cases. As seen in KR01, the 40-member ensembles used there achieve a good compromise between accuracy and keeping the cost of the data assimilation within acceptable limits (also see Section 3.3).

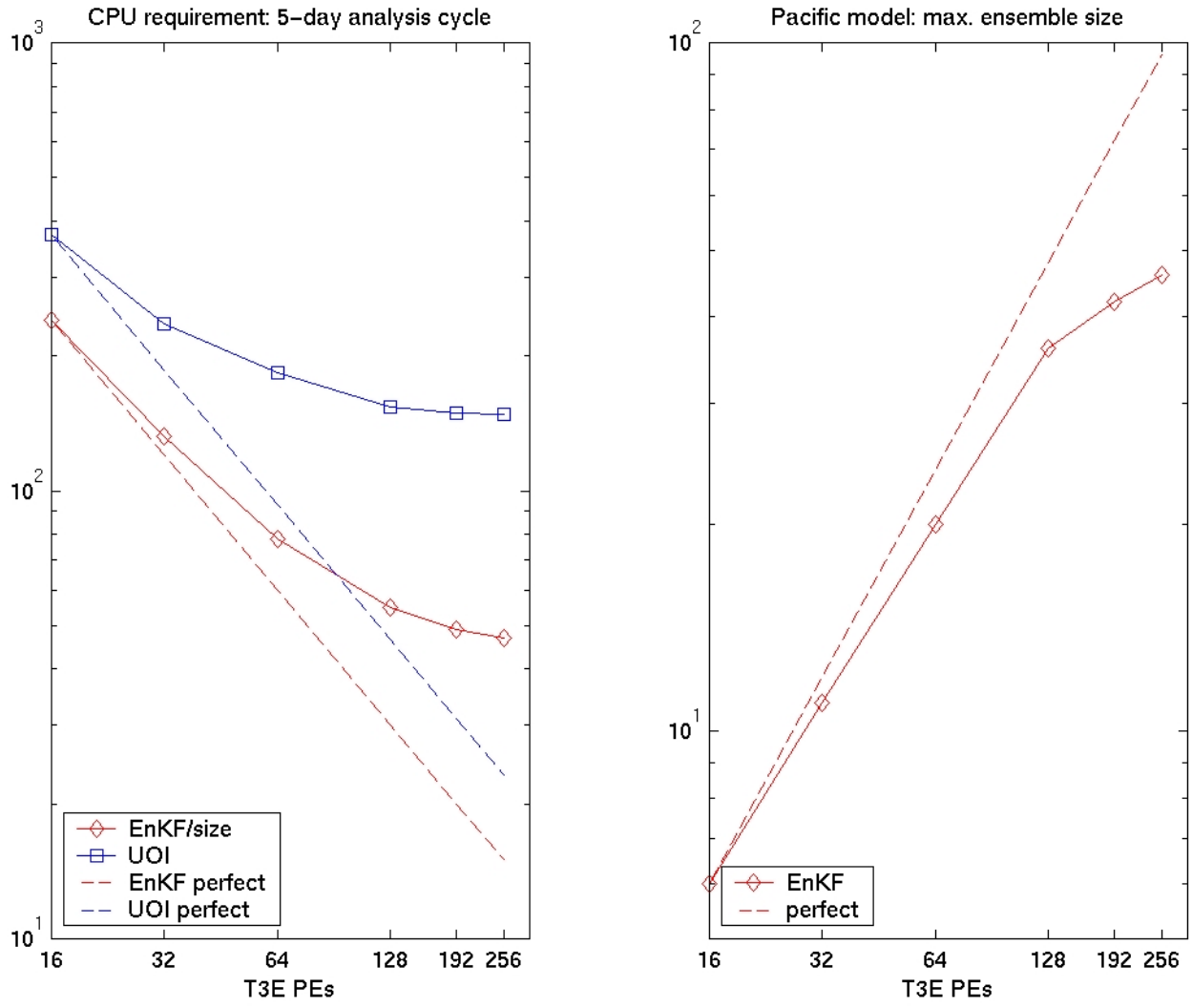

Figure 9. (a) Time per ensemble member required to complete one five-day analysis cycle when TAO temperature data are assimilated (t_m in text). The curves labeled "perfect" correspond to a unattainable linear scaling. (b) Largest ensemble size possible as a function of N_{PE} (m_{max} in text) on the CRAY T3E-600.

5 Summary

This article describes the MvEnKF design and its parallel implementation for the Poseidon OGCM. A domain decomposition whereby the memory of each PE contains that portion of every ensemble member's state vector which corresponds to the PE's position on a 2D horizontal lattice is used. The assimilation is parallelized through a localization of the forecast error-covariance matrix. When data become available to assimilate, each PE collects from neighboring PEs the innovations and measurement-functional elements according to the localization strategy. The covariance functions are given compact support by means of a

Hadamard product of the background-error covariance matrix with an idealized locally supported correlation function. In EnKF implementations involving low-resolution models, one has the freedom to work with ensemble sizes on the order of hundreds or thousands. Rather, with the state-vector size of approximately two million variables considered here, memory, interprocessor communications and operation count limit the ensemble size. Here, 40 ensemble members are used and the model domain is distributed over 256 CRAY T3E PEs.

Besides the details of the observing system implementation, the impact of the background-covariance localization on the analysis increments, as well as timing and scaling issues, were discussed. The validation of the MvEnKF in experiments involving TAO-temperature and TOPEX altimeter data is discussed in a companion article referred to herein as KR01.

Some issues that must be addressed to improve the MvEnKF are the deficiency of the system-noise model which only accounts for forcing errors, the problem of ensemble initialization which can be addressed using a perturbation-breeding approach, and the memory limitations inherent with running the MvEnKF on a MPP with distributed memory. On a machine with globally addressable memory, the memory-imposed constraints will be less severe. Fortunately, the modular, object oriented approach used to implement the MvDAS is not tied to the CRAY T3E architecture.

6 References

Atlas, R., R. Hoffman, S. Bloom, J. Jusem, and J. Ardizzone, 1996: A multiyear global surface wind velocity dataset using SSM/I wind observations. *Bull. Amer. Met. Soc.*, **77**, 869-882.

Bacmeister, J., and M. Suarez, 2001: Wind stress simulations and equatorial dynamics in an AGCM. Part I: Basic results from a 1997-1999 forced SST experiment, *J. Atmos. Sci*., submitted.

Bennett, A., 1992: *Inverse Methods in Physical Oceanography*. Cambridge University Press, 346pp.

Blanchet, I., and C. Frankignoul, 1997: A comparison of adaptive Kalman filters for a tropical Pacific Ocean model. *Mon. Wea. Rev.*, **125**, 40-58.

Bloom, S., L. Tacaks, A. DaSilva, and D. Ledvina, 1996: Data assimilation using incremental analysis updates. *Mon. Wea. Rev.*, **124**, 1256-1271.

Borovikov, A., M. Rienecker, and P. Schopf, 2001: Surface heat balance in the Equatorial Pacific Ocean: climatology and the warming event of 1994-95. *J. Clim.*, **14**, 2624-2641.

Burgers, G., P. van Leeuwen, and G. Evensen, 1998: Analysis scheme in the ensemble Kalman filter, *Mon. Wea. Rev.*, **126**, 1719-1724.

Cane, M., A.Kaplan, R. Miller, B. Tang, E. Hackert, and A. Busalacchi, 1996: Mapping Tropical Pacific sea level: data assimilation via a reduced state space Kalman filter. *J. Geophys. Res.*, **C101**, 22,599-22,617.

Cohn, S., A. da Silva, J. Guo, M. Sienkiewicz, and D. Larrich, 1998: Assessing the effect of data selection with the DAO Physical Space Statistical Analysis System. *Mon. Wea. Rev.*, **126**, 2913-2926.

Daley, R., 1991: *Atmospheric Data Analysis*. Cambridge University Press, 457pp.

Daley, R., and E. Barker, 2001: NAVDAS: formulation and diagnostics. *Mon. Wea. Rev.*, **129**, 869-883.

Dee, D. and A. Da Silva, 1998: Data assimilation in the presence of forecast bias. *Q. J. R. Meteorol. Soc.*, **124**, 269-295.

Evensen, G., 1994: Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *J. Geophys. Res.*, C**99**, 10,143-10,162.

Evensen, G., and P. van Leeuwen, 1996: Assimilation of GEOSAT altimeter data for the Agulhas current using the ensemble Kalman filter with a quasigeostrophic model. *Mon. Wea. Rev.*, **124**, 85-96.

Fukumori, I., and P. Malanotte-Rizzoli, 1995: An approximate Kalman filter for ocean data assimilation - an example with an idealized Gulf-Stream model. *J. Geophys Res.*, **C100**, 6777-6793.

Gaspari, G., and S. Cohn, 1999: Construction of correlation functions in two and three dimensions. *Quart. J. Roy. Meteor. Soc.*, **125**, 723-757.

Technical Staff, Analytical Science Corporations, 1974: *Applied Optimal Estimation*. A. Gelb, Ed., MIT Press, 374pp.

Ghil, M., and P. Malanotte-Rizzoli, 1991: Data assimilation in meteorology and oceanography. *Adv. Geophys.*, **33**, 141-266.

Hamming, R., 1983: *Digital Filters, 2nd edition*. Prentice Hall, 257pp.

Hamill, T., and C. Snyder, 2000: A hybrid ensemble Kalman filter-3D variational analysis scheme, *Mon. Wea. Rev.*, **128**, 2905-2919.

Horn, R., and C. Johnson, 1991: *Topics in Matrix Analysis*. Cambridge University Press, 615pp.

Houtekamer, P., and H. Mitchell, 1998: Data assimilation using an ensemble Kalman filter technique. *Mon. Wea. Rev.*, **126**, 796-811.

Houtekamer, P., and H. Mitchell, 2001: A sequential ensemble Kalman filter for atmospheric data assimilation. *Mon. Wea. Rev.*, **129**, 123-137.

Ji, M., and A. Leetmaa, 1997: Impact of data assimilation on ocean initialization and El Niño prediction, *Mon. Wea. Rev.*, **125**, 742-753.

Kalman, R., 1960: A new approach to linear filtering and prediction problems. *J. Basic Eng.*, **D82**, 35-45.

Keppenne, C., 2000: Data assimilation into a primitive-equation model with a parallel ensemble Kalman filter. *Mon. Wea. Rev.*, **128**, 1971-1981.

Keppenne, C., and M. Rienecker, 2001: Development of a parallel ensemble Kalman filter with the Poseidon ocean general circulation model, *Mon. Wea. Rev.*, submitted.

Konchady, M., A. Sood, and P. Schopf, 1998: Implementation and performance evaluation of a parallel ocean model. *Parallel Comput.*, **24**, 181-203.

Koster, R., and M. Suarez. 1996: The influence of land surface moisture on precipitation statistics, *J. Climate*, **9**, 2551-2567.

Knuth, D., 1998: *Sorting and Searching. The Art of Computer Programming*, Vol. 3, Addison-Wesley, 780pp.

Lermusiaux, P., and A. Robinson, 1999: Data assimilation via error subspace statistical estimation. Part I: theory and schemes. *Mon. Wea. Rev.*, **127**, 1385-1407.

Lorenc, A., 1981: A global three-dimensional multivariate statistical interpolation scheme, *Mon. Wea. Rev.*, **108**, 701-721.

McPhaden, M., A. Busalacchi, R. Cheney, J. Donguy, K. Gage, D. Halpern, M. Ji, P. Julian, G. Meyers, G. Mitchum, P. Niiler, J. Picaut, R. Reynolds, N. Smith, and K. Takeuchi, 1998: The Tropical Ocean-Global Atmosphere observing system: a decade of progress. *J. Geophys. Res.*, **C103**, 14169-14240.

Message Passing Interface Forum, 1994: A message-passing interface standard. CS 94-230, Computer Science Department Tech. Rep., University of Tennessee, Knoxville, TN, 228pp. [Availalble online at http://www.cs.utk.edu/~library/1994.html].

Mitchell, H., and P. Houtekamer, 2000: An adaptive ensemble Kalman filter. *Mon. Wea. Rev.*, **128**, 416-433.

Niiler, P., and E. Kraus, 1977: One-dimensional models of the upper ocean. *Modeling and Prediction of the Upper Layers of the Ocean*. E. Kraus, Ed., Pergamon, 143-172.

Pacanowski R., and S. Philander, 1981: Parameterization of vertical mixing in numerical models of the tropical oceans. *J. Phys. Oceanogr.*, **11**, 1443-1451.

Pham, D., J. Verron, M. Roubaud, 1998: A singular evolutive extended Kalman filter for data assimilation in oceanography. *J. Marine Sys.*, **16**, 323-340.

Reynolds, R., and T. Smith, 1994: Improved global sea-surface temperature analyses using optimum interpolation. *J. Climate*, **7**, 929-948.

Schaffer, D., and M. Suarez, 1998: Next stop: teraflop; the parallelization of an atmospheric general circulation model. draft manuscript, disponible in Postcript format from `http://nsipp.gsfc.nasa.gov/pubs.html`.

Schopf, P., and A. Loughe, 1995: A reduced-gravity isopycnic ocean model–hindcasts of El-Niño. *Mon. Wea. Rev.*, **123**, 2839-2863.

Suarez, M., and L. Takacs, 1995: *Documentation of the Aries/GEOS Dynamical Core: Version 2*. Technical Report Series on global modeling and data assimilation, Vol. 5, NASA Tech. Memorandum 104606, 45pp.

Verlaan, M., and A. Heemink, 1997: Tidal flow forecasting using reduced rank square root filters. *Stoch. Hydrol. Hydraul.*, **11**, 349-368.

Yang, S., K. Lau and P. Schopf, 1999: Sensitivity of the tropical Pacific Ocean to precipitation induced freshwater flux, *Clim. Dynam.*, 15, 737-750.

Yu, Z., P. Schopf and J. McCreary, 1997: On the annual cycle of upper-ocean circulation in the eastern Equatorial Pacific. *J. Phys. Oceanogr.*, **27**, 309-324.